

Home Work #2

Write a program in Python 3 (3.9 recommended) using no other library besides `Numpy` to implement a grading program as described below. You must make use of NumPy functions wherever possible.

- §1. The input consists of two CSV files. The data file, `data.csv`, contains a line of scores per student plus a few non-graded attributes like ID, attendance record, and the grade last assigned. A second file, `about-data.csv`, provides interpretation; it provides the column headers, the weights of each item (homework, exam, etc.), based on what possible maximum each score should be judged, the group each item belongs to, and one drop directive that specifies at most one group for which the lowest score should be dropped.
- §2. Your program must compute for each student an overall score for all items in groups numbered 1 or higher as well as the overall score separately for items in each of those groups.
- The overall score for some items is a weighted sum of the scores for those items weighted by the specified weights for those items. For example, if a student has a score of 12 in an item for which `OUTOF=30` and `WEIGHT=6`, then that student has earned 40% for this item ($= 100 \cdot 12 / 30$) and thus a contribution of 2.4 ($= 40\%$ of 6) towards the overall total. A missing score should be taken as zero for this computation.
 - (Extra Credit:) If a drop directive is present along with a group number, then the *lowest score* obtained in the items in that group would have to be dropped for each student; that item may be different for each student. If more than one item has the same lowest score, then the first should be chosen arbitrarily. You should check to ensure that the weights for each item in such a group are equal.
You should design your code so that it is easy to extend to enable this feature for more than one group.
If you are not aiming for extra credit, you can simply ignore the drop directive.
- §3. Use the `corrcoef()` function in NumPy to compute the Pearson Correlation coefficient between
1. *Attendance* and the overall score you have computed;
 2. *LastGrade* and the overall score you have computed; and
 3. *ID* and the overall score you have computed!
- §4. Your program should *neatly* print in tabular form the following.

1. Table 1: for each student, the ID, the overall score for each group numbered 1 or higher and finally the overall score.
It should have $g + 2$ columns if there are groups $0, 1, \dots, g$.
2. Table 2: For each item (homework 1, homework 2, etc.), the average, standard deviation, maximum and minimum for the entire class (in that order). However, you must *ignore* all missing entries.
It should have 5 columns.
3. Table 3 should report the results of 3.
It should have 3 rows and 2 columns. The first item in each row should indicate one of *Attendance*, *LastGrade*, *ID*; the second the Pearson correlation coefficient scores you have computed for this row item (see 3; follow that order).
4. (Extra Credit:) Output all the data in the input data file with one addition: add an extra column with the text 'Item ***' was ignored score (where *** is one of the items in the group with a drop directive).

§5. The input text file is presumed to be in the current directory.

§6. Use constants `DATAFILE` and `ABOUT_DATA` with string values equal to the given filenames.

§7. Your program should not crash even if, for example, the sum of the weights of a group is zero.

§8. Your program must be readable and understandable with appropriate decomposition and documentation. Constants other than the two mentioned above should be conspicuous, e.g., a `'Const_'` prefix.

The program should start with your name and outline the goal of the program and what it achieves (and list requirements that are not fully operational). You should announce whether or not you have tackled the extra credit option.

Each function should start with a docstring describing it and its inputs and be robust.

§9. Submit a Jupyter Notebook whose cells I can run in order. (Control the number of cells.)

Its last cell should contain a call to `main()`.